REMARKS/ARGUMENTS

I. Introduction:

Claims 1, 15, 19, and 20 are amended and new claims 22-34 are added herein. With entry of this amendment, claims 1-34 will be pending.

II. Claim Rejections under 35 U.S.C. 102 and 103:

Claims 1, 3, 4, 8-13, 15, 16, and 18-20 stand rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 5,265,092 (Soloway et al.).

Claim 1 is directed to a method for performing route calculations in a link state routing protocol at a node within a computer network. The method includes receiving new route information at the node, evaluating existing routes of the node, and recalculating routes for the node only when the new route information improves at least one of the existing routes or at least one of the existing routes is made worse or lost. Claim 1 has been amended to clarify that the routes are evaluated to determine if new route information improves at least one of the existing routes or at least one of the existing routes is made worse or lost, and that the node's routing table is modified when the routes are recalculated.

The Soloway et al. patent is directed to a synchronization mechanism for link state packet routing. A system and method for providing loop free and shortest path routing of data packets are disclosed. The routing of data packets is achieved through modifications to conventional link state packet (LSP) routing protocols and permits each switch to inform adjacent switches in the network of information in the switch's database used to compute forwarding tables. A switch uses a received LSP to compute a forwarding table and informs neighboring switches on attached links of the routing change. The switches determine the shortest paths for constructing forwarding tables using a shortest path calculation algorithm such as Dijkstra's algorithm. The switches compute shortest paths from each attached channel's link to all end nodes and use the results of these calculations to construct their forwarding tables. After receiving a routing change, the switch discards any subsequent data packets whose path would be affected by the changed routing information. Shortest path routing for data packets from a source endnote to a destination end node is achieved by assuring that the first switch to forward the packet is on the shortest path to the packet's destination endnote.

Soloway et al. do not disclose recalculating routes and modifying a routing table for a node only when new route information improves at least one of the existing routes or at least one of the existing routes is made worse or lost. The system of Soloway et al. uses a received LSP to compute a forwarding table (routing table). The shortest path is found by performing a shortest path calculation and creating a new forwarding table once the LSP is received with new routing information. Soloway et al. do not determine whether the new route information improves a route or makes an existing route worst or lost before recalculating routes and creating a new forwarding table. In rejecting claim 1 the Examiner refers to col. 3, line 13-24 and col. 4, lines 19-65. Col. 3 discusses how the system achieves loop-free routing of data packets through modifications to conventional LSP routing protocols. When a switch first uses a received LSP to compute a forwarding table it discards any data packet that the switch receives whose path would be affected by the new routing information. The switch continues to discard the affected data packets until the switch receives notification from each adjacent switch affected by the new routing information that all affected routing paths have been re-calculated and the forwarding tables have been updated. Thus, the switch discards certain packets after receiving an LSP with new route information to maintain loop-free data packet routing until all of the forwarding tables have been updated. Col. 4 discusses how the protocol assures that no two switches on the same link having different switch lists will consider the link up and that the routing logic will not receive and process a packet sent by a switch before a link state changes from up to down. There is no discussion in Soloway et al. of evaluating existing routes to determine if new route information improves at least one of the existing routes or at least one of the existing routes is made worse or lost before recalculating routes to create a new forwarding table.

Accordingly, claim 1 is submitted as not anticipated by Soloway et al. and the other prior art of record.

Claims 2-12 and new claims 29-33, depending either directly or indirectly from claim 1, are submitted as patentable for the same reasons as claim 1.

Claim 15 is directed to a computer program product for performing route calculations in a link state routing protocol and claims 19 and 20 are directed to systems for performing route calculations in a link state routing protocol. Claims 15, 19, and 20 have also been amended to clarify that the routes are evaluated to determine if new route information improves at least one of the existing routes or at least one of the existing routes is made worse or lost, and that the node's routing table is modified when the routes are recalculated. Claims 15, 19, and 20, and claims 16-17, depending directly from claim 15, are submitted as patentable for the reasons discussed above with respect to claim 1.

Claim 13 is directed to a method for updating a tree structure of a root node in a computer network of interconnected nodes after a change in the network's topology. The method includes receiving new route information at the root node and applying an incremental Dijkstra's algorithm to the root node only if the new route information improves or worsens at least one of the existing routes or at least one of the existing routes is lost.

Applicant respectfully submits that claim 13 is not anticipated by Soloway et al. As previously discussed, Soloway et al. create a new forwarding table when new route information is received and the table is constructed using a conventional Dijkstra's algorithm. In contrast, applicant's invention, as set forth in claim 13, uses an incremental Dijkstra's algorithm applied to the root node only if new route information improves or worsens at least one of the existing routes or at least one of the existing routes is lost. The conventional implementation of the Dijkstra algorithm requires that every node in the network be examined for every network topology change. Applicant's invention is particularly advantageous in that the number of nodes to be examined is reduced so that the computation required is only a fraction of the conventional implementation.

Accordingly, claim 13 is submitted as not anticipated by Soloway et al. and the other prior art of record.

Claims 14 and 34, depending directly from claim 13, are submitted as patentable for the same reasons as claim 13.

Claim 18 is directed to a computer program product for updating a tree structure of a root node in a computer network of interconnected nodes after a change in the network's topology and is submitted as patentable for the reasons discussed above with respect to claim 13.

The additional references cited including U.S. Patent No. 6,456,599 (Elliott), do not remedy the deficiencies of the primary reference.

III. Conclusion:

For the foregoing reasons, Applicant believes that all of the pending claims are in condition for allowance and should be passed to issue. If the Examiner feels that a telephone conference would in any way expedite the prosecution of the application, please do not hesitate to call the undersigned at (408) 446-8695.

Respectfully submitted,

Cindy S. Kaplan
Reg. No. 40,043

RITTER, LANG & KAPLAN LLP
12930 Saratoga Ave., Suite D1
Saratoga, CA 95070
Tel: 408-446-8690/ Fax: 408-446-8691